HW7

黄佳溢

December 24, 2024

1 Python生成及搬运

这个部分在课件上也已经说明详尽了，只需要依样画葫芦即可，这里仅点出需要注意的几个事项;

1. **加入龙芯支持项**

   我们在执行configure配置时总会出现"**mechine 'loongarch64' not recognized**"，这是因为工具库源码并不支持龙芯架构，我们只需要为其添加支持权限即可。具体操作为**寻找源码中的config.sub文件，输入"/powerpc"可以跳转到支持架构的位置，在那里加入"loongarch64"即可**。

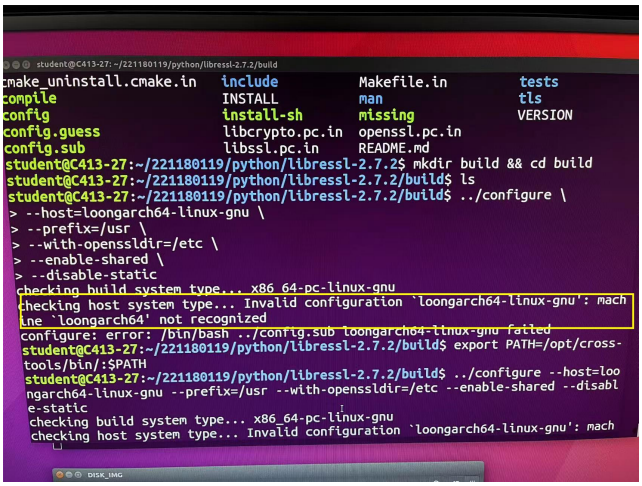

Figure 1: 报错

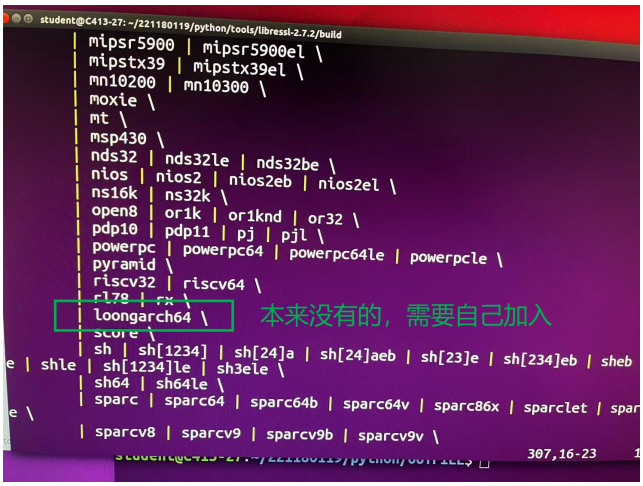

Figure 2: 修改

还需要注意的是，① 为了方便寻找config.sub的位置，我们可以调用find -name config.sub指令来加快效率； ② 在config.sub文件中，有时候输入powerpc也不一定能直接

跳转到具体的位置，此时可以按n键来跳转到下一个位置[1]；

2. **无法搬运**

这种情况常常出现在，在嵌入式开发板中，将/target/usr/bin目录下的东西复制到/usr/bin目录下的过程，具体报错见图

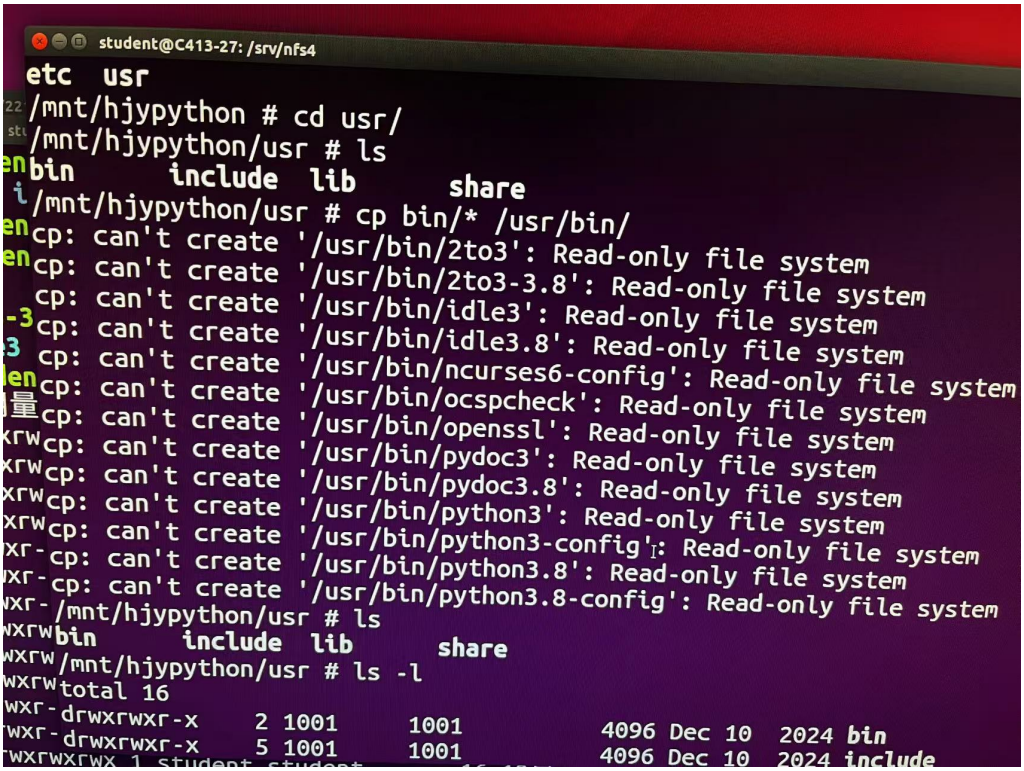

Figure 3: Read-only file system

这是由于我们在BootLoader的交互式启动环节，启动的代码为：

g console=ttyS0, 115200 root=/dev/ram0

这表明嵌入式系统将默认为**仅写入**的权限，所以我们就无法对其进行读写操作 解决方法就是在启动的时候，在后面加入**rw**。

g console=ttyS0, 115200 root=/dev/ram0 rw

3. **没内存**

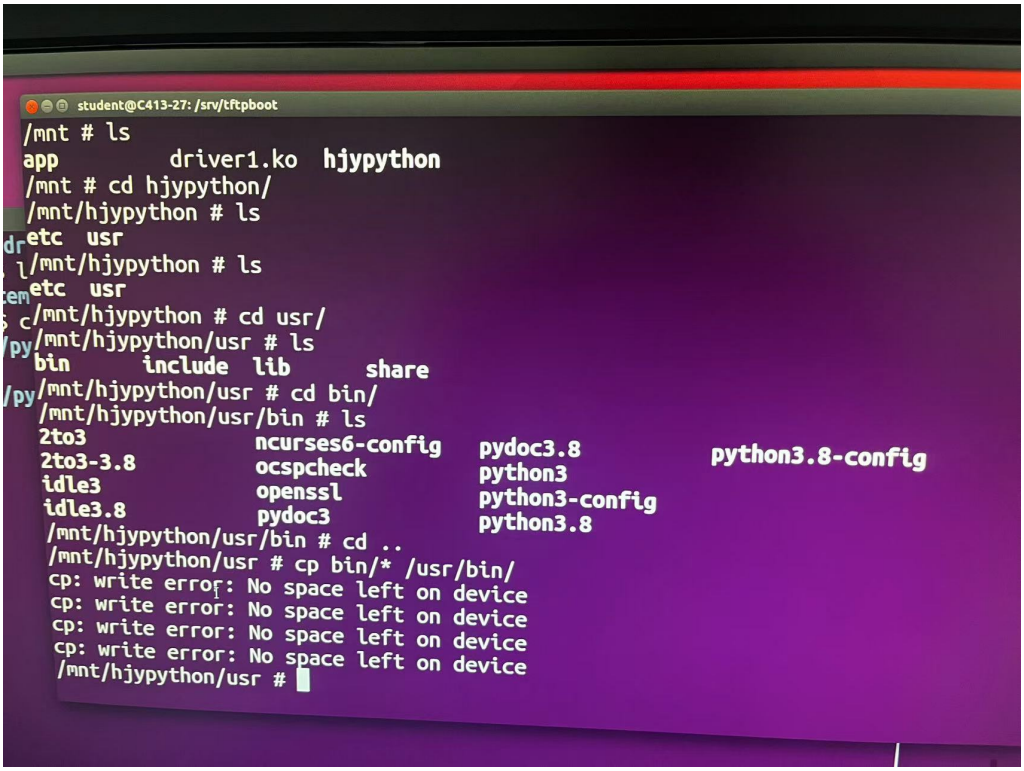

Figure 4: No space left on device

这个问题因人而异，有些人在配置文件系统时，默认的给定的设备空间比较大，所以不会出现这个问题（感谢前辈）。我比较倒霉，遇到的设备空间比较小QAQ， 这是一个很难绷的问题，首先我需要解释这个现象出现的原因：

① 在嵌入式开发板中，除了挂载的mnt外，其他的文件目录都是**占用了开发板自己的硬盘空间**，在文件系统配置比较小的时候，加上提前创建好的其他文件，剩余空间已经岌岌可危，亲测剩下的空间仅能装入几个.c文件，但是对于python这种大可执行文件而言，是完全无法转移的。

② 挂载成功之后的mnt目录比较特殊，它的底层是NFS（网络文件系统），是我们的主机提供给开发板的共享文件夹，所使用的是**主机的空间**，所以我们在开发板终端下，在mnt目录做的一切操作都是在**主机空间**里进行，所以不会出现空间不足的问题。

为解决这个问题，这里我给出两种思路：①（正规做法）重装文件系统，将分配空间做大； ②（非正规做法）将所有的环境变量都指向mnt目录，在mnt目录下操作。这里我选择**第二种方案**（嘻嘻）并在下一个章节为你解释这种做法的可行性。

2 环境配置

2.1 原理解释

1. **PATH**

   系统用于查找**可执行文件或指令**位置的全局变量，我们熟知的ls、pwd等指令都需要在PATH中点明路径，通常位于/usr/bin目录下。 如果我们希望在任意路径下都能使用python3.8来进入交互界面，我们也需要在PATH中加入python3.8所在的位置。

   export PATH=%python3.8所在的位置%:$PATH

2. **LD_LIBRARY_PATH**

   当系统执行可执行文件时，总会引用一些动态库，此时需要指出动态库存在的位置。而LD_LIBRARY_PATH就是记录这些路径的全局变量，所以在执行python3.8时，需要指出它的动态库位置:

   export LD_LIBRARY_PATH=%python3.8动态库所在位置%:$LD_LIBRARY_PATH

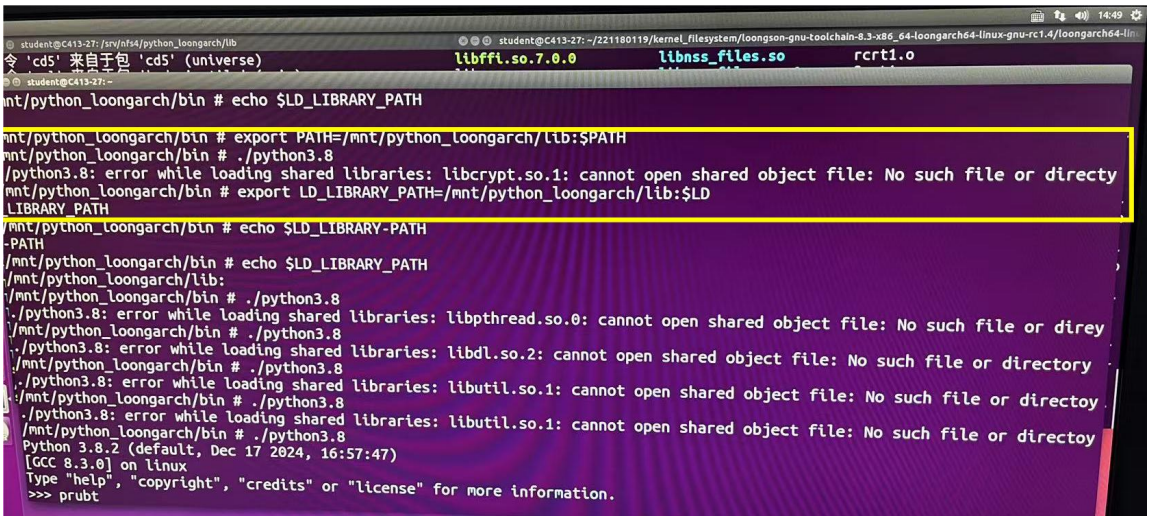

Figure 5: 设置全局环境变量

2.2 调试

接下来的操作还是很自然了，我们只需要不断输入**python3.8**，然后看报错还差什么动态库就去**主机**上寻找，复制到**lib**路径下。比如，当缺失**libpthread.so.0**时，可以切换到主机的根目录，执行**find -name libpthread.so.0**进行查找。

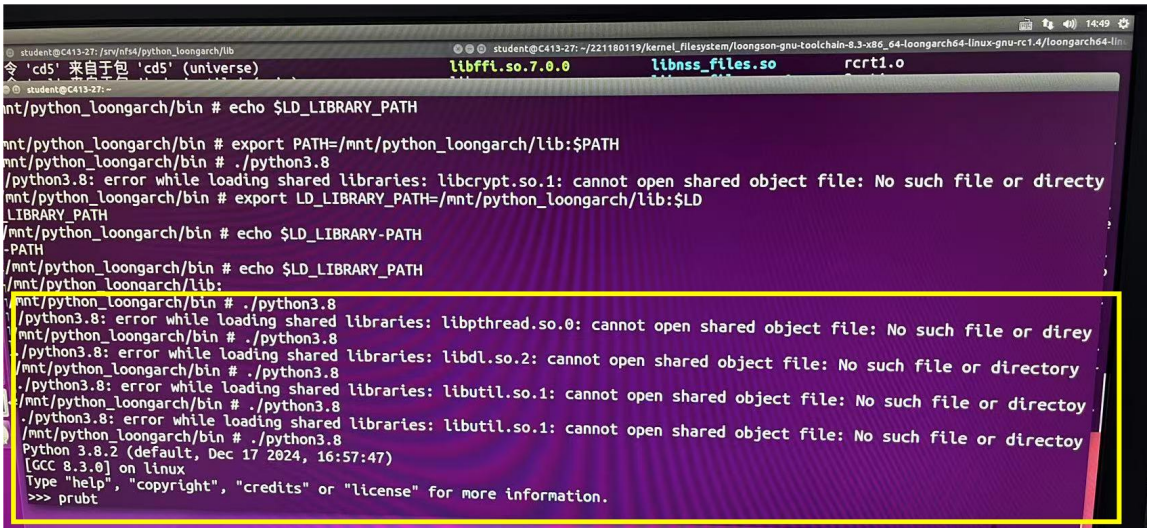

Figure 6: 调试

2.3 实现效果

下面展示了交互界面运行和.py文件执行的效果

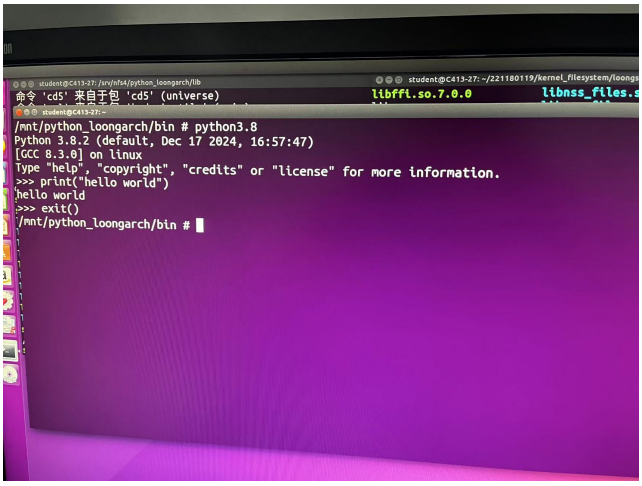

Figure 7: 交互界面运行效果

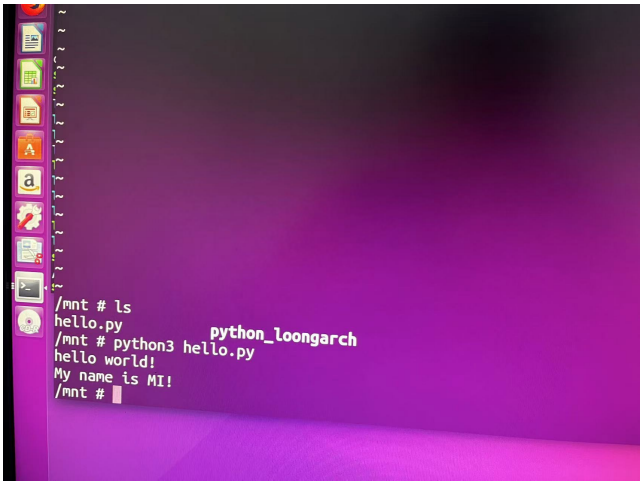

Figure 8: 运行.py文件效果

3 总结

本学期的数字系统II实验到这里也就圆满结束了，在本课程中，我深入学习了嵌入式系统的运行原理，为我将来的科研打下了坚实的基础（撒花）在这里还是感谢所有为我提供帮助的老师，助教和同学！

唯一可惜的是，由于实验的文件体积较大（达到了6.3GB）所以无法将我的全部代码进行开源QAQ，但是我还是会将所有的实验报告上传到Github仓库！为南京大学的开源社区贡献一份力量！（耶）